**Assignment 2: Monitors and Semaphores**

Asher Shores, Philip Varkey

College of Science, Engineering, and Technology, Grand Canyon University

CST-315: Operating Systems Lecture & Lab

Dr. Ricardo Citro

13 February 2022

**Activity Directions:**

In this project, students will demonstrate an understanding of the difference between *monitors* and *semaphores*. Define a simple scenario (obviously in the context of processes and threads in Unix/Linux) in which both monitors and semaphores could be used. Implement your scenario in two separate C programs, one using semaphores and the other using monitors.

**Scenario**

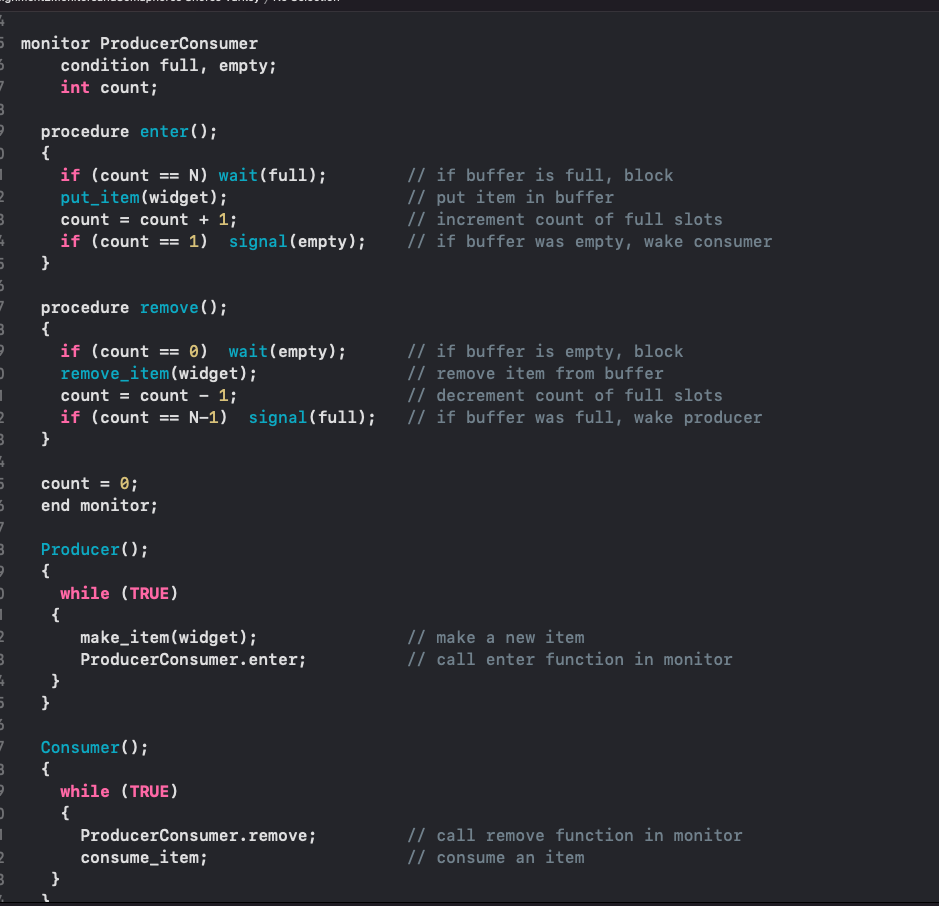
The most appropriate scenario for using a monitor and/or semaphore approach is the classic Consumer-Producer Problem. This historic problem is often cited in multi-process synchronization where a producer produces tasks, and the consumer consumes those tasks. Both threads user the same memory buffer. The producer creates data to place in the buffer from which the consumer consumes from. The goal is to find balance between the two without creating deadlock. The producer should only work while the buffer is not full and the consumer only when the buffer is not empty, and either cannot use the buffer while the other currently is.

**Explanation and Justification**

One solution to the problem described above is the use of monitors which make organizing the system easier. Monitors allow for mutual exclusion by placing the so-called “critical section” of the program within the monitor. This allows the process to be controlled, via wait and signal primitives. The other solution involves using semaphores. Using specifically counting semaphores allows us to better track the buffer at all times. Using two of these counting semaphores allows us to track the number of items in the buffer as well as the amount of free space with the other. Thus, we can actively control the access of the resource (buffer) using the semaphores.

**Code Snippets**

**Monitor Code**

****

**Semaphore Code**

**Semaphore Code Execution**

**Text

Description automatically generated with low confidence**

**GitHub Link:**

**https://github.com/asherShores5/Assignment-2-Monitors-and-Semaphores**

**References**

GeeksforGeeks. (2021, September 20). *Producer Consumer Problem using Semaphores | Set 1*. Retrieved February 13, 2022, from https://www.geeksforgeeks.org/producer-consumer-problem-using-semaphores-set-1/

*The Producer-Consumer problem in Operating System*. (2019, November 16). AfterAcademy. Retrieved February 13, 2022, from https://afteracademy.com/blog/the-producer-consumer-problem-in-operating-system#:%7E:text=The%20Producer%2DConsumer%20problem%20is,products%20produced%20by%20the%20Producer.